INDOOR LIDAR RELOCALIZATION BASED ON DEEP LEARNING USING A 3D MODEL

H. Zhao[1*], D. Acharya[1], M. Tomko[2], K. Khoshelham [2]

[1]Dept. Infrastructure Engineering, The University of Melbourne, Australia - (zhaohz, acharyad) @student.unimelb.edu.au
[2]Dept. Infrastructure Engineering, The University of Melbourne, Australia - (k.khoshelham, tomkom) @unimelb.edu.au

KEY WORDS: Relocalization, CNN pose regression, 3D model, Synthetic image, Point cloud, Sensor pose

ABSTRACT:

Indoor localization, navigation and mapping systems highly rely on the initial sensor pose information to achieve a high accuracy. Most existing indoor mapping and navigation systems cannot initialize the sensor poses automatically and consequently these systems cannot perform relocalization and recover from a pose estimation failure. For most indoor environments, a map or a 3D model is often available, and can provide useful information for relocalization. This paper presents a novel relocalization method for lidar sensors in indoor environments to estimate the initial lidar pose using a CNN pose regression network trained using a 3D model. A set of synthetic lidar frames are generated from the 3D model with known poses. Each lidar range image is a one-channel range image, used to train the CNN pose regression network from scratch to predict the initial sensor location and orientation. The CNN regression network trained by synthetic range images is used to estimate the poses of the lidar using real range images captured in the indoor environment. The results show that the proposed CNN regression network can learn from synthetic lidar data and estimate the pose of real lidar data with an accuracy of 1.9 m and 8.7 degrees.

1. INTRODUCTION

Lidar SLAM (Simultaneous Localization and Mapping) has been widely studied in recent decades for data collection and mapping in indoor environments. Lidar sensors provide rich distance measurements which can be converted to a point cloud providing an accurate representation of indoor structural features (Cheng *et al.*, 2018). Existing SLAM algorithms highly rely on the initial pose of the sensor to be able to recover from possible pose estimation failures. If the localization and orientation estimation algorithm performs poorly, it will be necessary for the algorithms to relocalize the sensor and recover the location and orientation from the failure.

SLAM algorithms localize the sensor and map the environment incrementally by estimating the sensor pose with respect to a previous pose. If this incremental localization fails, the algorithm needs to re-estimate the sensor pose with respect to the map without using previous pose estimates. This is referred to as relocalization.

Relocalization algorithms initialize the sensor pose by using an existing map such as a point cloud captured previously by the sensor, or an existing 3D model or a floor plan (Caron *et al.*, 2014). The existing map can provide useful information to help estimate the sensor pose. When the sensor used to acquire data needs to be relocalized automatically, the collected data can be matched with the existing map to estimate the location and orientation of the sensor (Wang *et al.*, 2017).

The relocalization methods can be mainly divided into two categories: vision-based methods and lidar-based methods (Shotton *et al.*, 2013; Kendall *et al.*, 2015; Wang *et al.*, 2017). Cameras provide rich visual information in indoor environments, which can help estimate the camera pose (Kendall *et al.*, 2015; Acharya *et al.*, 2019a). However, cameras are susceptible to texture and lighting conditions, and the acquired images may be blurred due to the camera motion or lack sufficient texture or brightness because of poor texture and lighting conditions. Compared with vision-based relocalization methods, lidar relocalization takes advantage of accurate range data, long range, and in wide field of view of 360 degrees, without artefacts such as image blur. Another advantage of lidar is that it is independent of light conditions and performs well in poorly textured environments (Saeedi *et al.*, 2016).

Conventional lidar relocalization methods estimate the pose by using a map previously captured by the sensor (Wang *et al.*, 2017; Tian *et al.*, 2019). This poses a practical challenge since the map generation in the first place depends on the relocalization ability to recover from possible failures.

In this paper, we propose a new lidar relocalization method based on an existing 3D model of the indoor environment which is often available or can be easily created from a floor plan. The contributions of this paper are the followings:

(1) We present a novel pose estimation method using lidar data and a 3D model. We generate a set of synthetic range images using the 3D model. These synthetic range images are used to train a CNN regression network. Each synthetic range image is associated with a position and an orientation in the real building provided that the 3D model represents the building accurately.

(2) We show that the CNN regression network trained by synthetic range images generated from the 3D model can accurately estimate the pose of real range images captured by the lidar sensor in the indoor environment.

2. RELATED WORK

If an existing map of the environment is available, the pose estimation system can rely on the existing maps to achieve a higher accuracy. With an existing map, localizing the robot problem is called relocalization.

* Corresponding author

Relocalizing the robot by matching the current frame image with previous images has been proposed by Reitmayr and Drummond. (2006). If the matching was successful, the pose of the camera could be recovered. If the current frame could not match well with previous images, the next image frame would be used for recovering. Tracking the interest objectives to relocalize the sensor has been proposed by Özuysal *et al.* (2006). They used classification instead of feature extraction and feature matching to track the objective. They trained a classifier to classify the input image as similar to one of the previously seen images, so the system could detect the re-occurrence even if the input image was blurred or noisy. Özuysal's system was improved by considering each class's score independently and the classifier returned all classes scoring higher than a threshold (Williams *et al.*, 2007). To relocalize the robot from a failure with the training set of images, they implemented an off-the-shelf method: using three feature correspondences and their three-point pose algorithm was provided by Fischler and Bolles (1981). Willianms's system provided the distribution and uncertainty of the pose, so feature correspondences were filtered by potential visibility. Thus, this system could implement relocalization immediately when the tracking algorithm was lost. The established map could be used to detect the re-occurrence of a place to reduce the drift of the trajectory (Mur-Artal and Tardós, 2017).

Matching RGB-D and depth information of an input image with an existing database could provide the pose estimate of the camera and compute the coordinates in the scene coordinate system (Shotton *et al.*, 2013). The existing database was a trained regression forest consisting of labeled pixels by calibrating depth information of pixels. The current image was input into the well-trained forest and an estimated camera pose would be outputted. Convolutional networks (CNN) also provided a powerful tool to perform the regression task by using images (Kendall *et al.*, 2015). In Kendall's system, the acquired real images were input into a convolutional network for training and the current image was input into the well-trained convolutional network for a real time 6-DOF camera pose estimation.

Model based approaches have also been explored by researchers in recent years. In model based approaches, the pose of a given image is computed by minimizing the error between measurements in the image and the projection of a 3D model of the scene (Caron *et al.*, 2014). A real-time SLAM algorithm has also been provided by Caron *et al.* (2014). Caron's approach combined the vision information with a 3D model. This system will extract the segment features of the image and match these segment features with the 3D model to estimate the pose of the robot and the pose estimation and optimization are obtained by implementing UKF. A camera pose estimation approach using a 3D model and deep learning networks has been proposed by Acharya *et al.* (2019a). A Bayesian and a recurrent network were used to train generated synthetic images using a 3D model, and the pose of the current image frame was estimated with the well-trained network (Acharya *et al.*, 2019b).

An relocalization algorithm with an existing 3D map has been proposed by Wang *et al.* (2017). With an established 3D map, 2D maps were sampled from this 3D map and the 2D observation was matched with these 2D maps to initialize the robot by implementing a particle filter. After estimating the initial pose of the robot, the extracted 2D point cloud from the current frame was matched with the extracted 2D maps from the existing 3D map to estimate the trajectory of the robot. A probabilistically sound method for relocalization is proposed by using scan-based maps for autonomous navigation (Schiotka *et al.*, 2017). They built a regular map with a known pose and known measurement scenario and then the Bayes filter theory was used for localization with the existing map by finding the best match by finding the minimal distance between the end point of the beam and the points in one scan. For establishing the map, they considered three strategies: selecting scans equidistantly along the trajectory, grouping poses with similar observations and finding the set of scans with the maximal observation probability. A relocalization algorithm has provided by Tian *et al.* (2019). Tian's system subdivided a 3D scene map into three parts evenly and vertically, and extracted the most informative point cloud layer for localization estimation. The current input lidar point cloud would be matched with the three parts of the existing 3D map independently using Normal Distributions Transform (NDT) algorithm (Zhang *et al.*, 2014). For pose estimation, the consistency detection of three poses were evaluated and barycenter or midpoint of the three results would be calculated with three weights computed by considering NDT score. To deal with the sparse gradient problem of the occupancy grid map, a relocalization method has been designed to covert the original occupancy grid map into ESDF (Euclidean SDF) and TSDF (Truncated SDF) (Zhang *et al.*, 2019). The distance difference of scan-scan constraints (difference between the current scan and previous scans) and the distance difference of scan-map constraints (difference between current scan and the existing map) were considered by implementing a sliding window algorithm. For mapping, each lidar point was projected to its corresponding grid cell and all the points were grouped to update the map. A data-driven descriptor training method was designed for relocalization and map reconstration (Dubé *et al.*, 2018). In Dubé's system, the current input lidar point segment features were extracted and this meant each frame of lidar point cloud corresponded to one set of segment features. The global map was established by accumulating centroids and descriptors of these extracted features and then this global map was used for relocalization: the local segments would be matched with the global map by KNN to find the best corresponds and this means the relocalization could be achieved by verifying the correspondences between the current point cloud and the geometric consistency.

3. LIDAR POSE ESTIMATION

3.1 Framework of the Lidar Pose Estimation

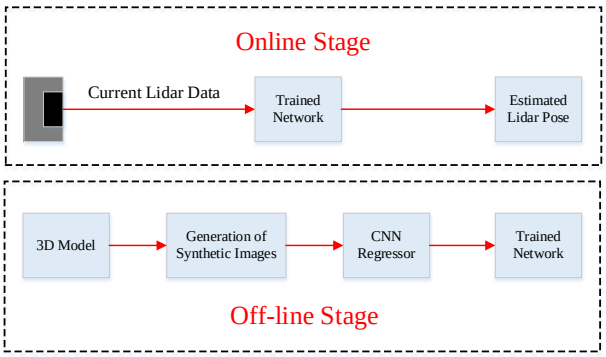

Figure 1. Workflow of our relocalization method

The proposed relocalization method can be divided into two stages: an online stage and an off-line stage, as shown in Figure 1. In the off-line stage, we simulate a lidar sensor placed in the 3D model to generate synthetic lidar range images with known

poses. These synthetic range images are then used to train a CNN regression network to train it. In the online stage, the current frame of lidar, i.e. a real range image, is input into the trained CNN regression network to estimate the real-time sensor pose with respect to the coordinate system of the 3D model.

3.2 Generation of Synthetic Images

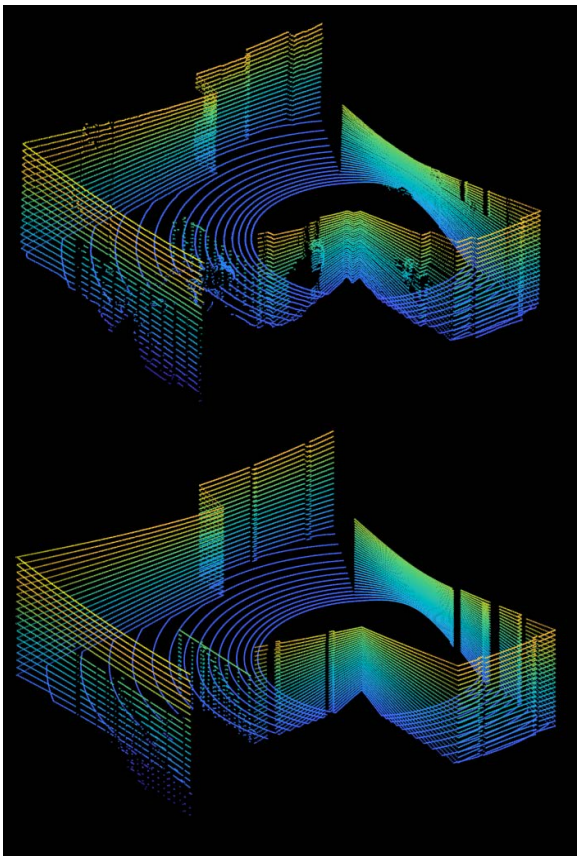

Figure 2. A real lidar point cloud (top) and the corresponding synthetic point cloud (bottom)

To achieve high pose estimation accuracy, the generated synthetic images should be similar to real lidar range images. In the paper, we implement a ray-tracing algorithm to generate synthetic range images from the 3D model. The real lidar dataset is acquired by a 32-channel Velodyne lidar, which can acquire range data in 32 certain vertical angles. For each channel, range data are acquired in approximate 2170 horizontal angles, for one complete rotation. To generate synthetic range images similar to real range images, a simulated lidar is placed in the 3D model and it fires rays in 32 vertical angels and 2170 horizontal angles. Each ray intersects with planes in the 3D model providing range values between each intersection and the origin point. The minimum range value is selected and stored in the corresponding location in the synthetic range image. Figure 2 shows a real lidar point cloud and a synthetic point cloud generated at the same location.

The generated synthetic range images are similar to the real range image but there are still some differences, such as missing points in the real images, caused by the transparent or specular surfaces. Compared with the real images, the synthetic images contain less details as the 3D model is by defining a simplified representation of the real environment.

3.3 CNN Regression Network Architecture

The generated synthetic images with known poses will be input into a CNN regression network for training and testing and thus, the architecture of the network used for training and testing has an impact on the levels of accuracy that can be achieved. Classical architectures can achieve high accuracy for image classification task (He *et al.*, 2016; Szegedy *et al.*, 2017). However, the synthetic range image is in the 32 x 2170 shape. In this paper, synthetic images are trained with a VGG-based CNN architecture and a ResNet-based architecture. Figure 3 shows the structure of ResNet-based architecture:

Figure 3. The structure of ResNet-based CNN architecture

As figure 3 shows, we replaced the average pooling and the following full connection layer with two full connection layers to perform the regression task.

3.4 Loss Function

To train the CNN regression network, we define the loss function as the pose estimation error. A 6-DOF pose describes the sensor location and sensor orientation. The sensor orientation is represented by quaternions. Expressing the sensor orientation using quaternions can achieve a higher accuracy, and training

location and orientation together performs better than training them separately (Kendall *et al.*, 2015; Kendall and Cipolla, 2017). The loss function is defined as follows:

$$L = \|(p-\hat{p})\|_2 + \beta\|(q-\hat{q})\|_2 \quad (1)$$

Where L = loss
p = location
q = quaternion
β = a weight parameter

p and q are the ground truth position and quaternion vector respectively. $\hat{p}$ and $\hat{q}$ are estimated position and quaternion vector respectively and $\|.\|_2$ denotes the L_2 norm.

In the equation above, the weight parameter β is used to balance the location loss and orientation loss. A unit quaternion ranges from 0 to 1 but the location can range from 0 to tens of meters (depending on the environment and the maximum range of the sensor), and this will let the training process focus more on location loss. With the weight parameter β, the training process will balance between the location loss and the quaternion loss.

4. EXPERIMENTS AND RESULTS

4.1 Training Set and Test Set Generation

The experiments were carried out by simulating a 32-channel Velodyne lidar scanner in a 3D model of a university building obtained from a public dataset (Khoshelham *et al.*, 2017). We placed a simulated lidar sensor at 0.35-meter distance intervals and 10-degree orientation intervals resulting in 231 positions and 36 orientations for each position. Each simulated lidar frame is a synthetic range image registered with a known pose. The whole set of synthetic ranges was used to train the CNN regression network. A test dataset was generated by placing the simulated lidar in random positions and random orientations in the 3D model. A set of real range images was also acquired by a 32-channel Velodyne lidar mounted on an unmanned ground vehicle in the real indoor environment. The vehicle Husky with the Velodyne lidar on it is shown in figure 4.

Figure 4. The Husky vehicle and Velodyne lidar

Figure 5 shows the generation of the training dataset and the test dataset using the 3D model. The map in yellow is top view of the 3D model of the test environment. The red arrows are the training lidar frames, with origin points representing the locations and the arrows representing the directions. We selected 231 locations uniformly distributed in the corridor and at each location, we generated 36 lidar frames in 36 directions at 10-degree intervals. In order to avoid using 231 fixed locations to estimate the sensor pose in the whole building, we added a Gaussian noise to the locations and orientations for each lidar frame in the training set. The blue arrows are the test lidar frames. We generated in total 8316 synthetic images for training and 924 synthetic range images for testing with random locations and random directions. 832 real lidar range images were acquired using the velodyne lidar for testing the trained CNN regression network.

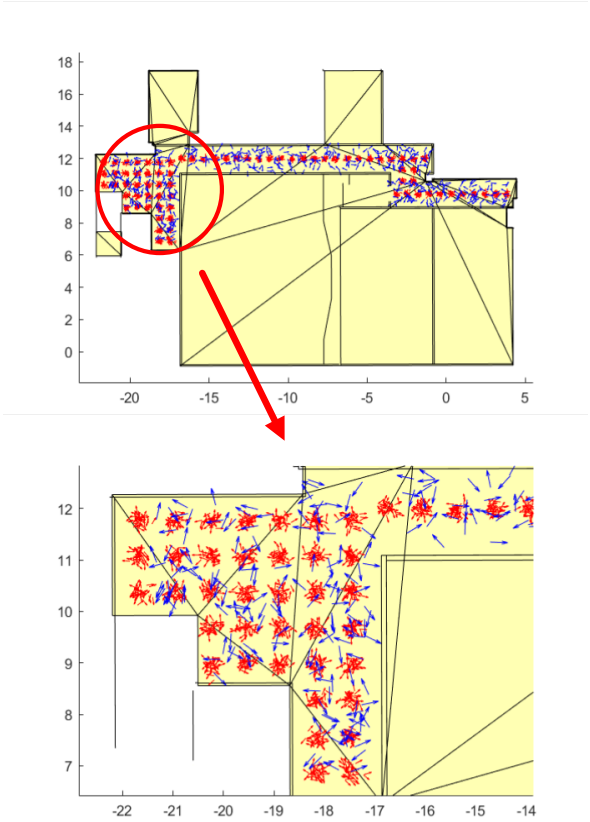

Figure 5. The generated training and test synthetic lidar frames

4.2 Training and Testing with Two CNN Regression Architectures

The generated synthetic training lidar frames with known poses are input into the VGG-based CNN regression architecture and the ResNet-based architecture for training and validation. Then the synthetic test dataset is used to evaluate the trained architectures.

Selecting a suitable β value is important to the training and validation process. If a large β value is selected, the training process will focus more on the quaternion loss. This means the location loss will decrease very slow while the quaternion loss will reach a low value and keep stable, and vice versa.

An appropriate value for the weight parameter β value is found empirically by conducting experiments until the trained architecture can achieve a satisfied accuracy.

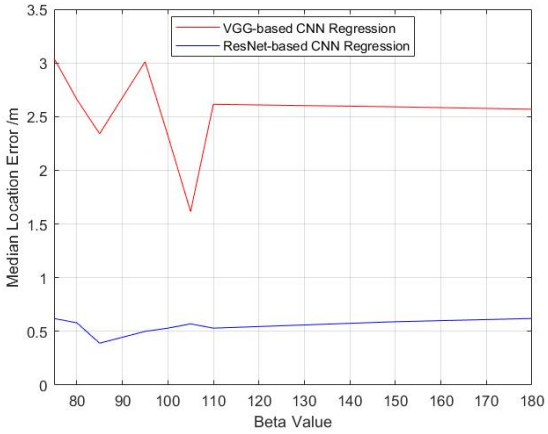

Figure 6. Median Positioning error with different β values

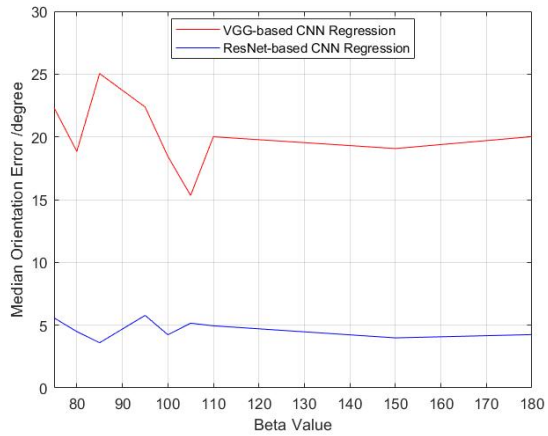

Figure 7. Median orientation error with different β values

Figure 6 and figure 7 show that selecting different β values achieves different level accuracy by testing the trained VGG-based CNN architecture and the ResNet-based architecture.

we find that a suitable β value for the weight parameter is 105 for the VGG-based regression architecture and 85 for the ResNet-based regression architecture. With the well-trained CNN regression architectures, a set of real lidar dataset acquired by the 32-channel Velodyne lidar will be input into this well-trained architecture to estimate the sensor pose. Table 1 shows the best median location accuracy and median orientation accuracy we have achieved and the corresponding β value with synthetic range images. The results in table 1 show that the VGG-based CNN regression network and the ResNet-based CNN regression network can learn from synthetic lidar data and estimate the pose of synthetic lidar data with the accuracy of 1.65 m and 15.6 degrees, and 0.39 m and 3.6 degrees respectively.

Architecture	Median Error	β value
VGG-Based	(1.65/m, 15.6/degree)	105
ResNet-Based	(0.39/m, 3.6/degree)	85

Table 1. Test results with synthetic range images

Now we have taken the networks trained on synthetic range images and the achieved a satisfied test accuracy. Then we apply the trained CNN regression networks to real range images.

Table 2 shows the best median location accuracy and median orientation accuracy we have achieved and the corresponding β value with real range images. The results in table 2 show that the VGG-based CNN regression network and the ResNet-based CNN regression network estimate the pose of real lidar data with the accuracy of 3.1 m and 18.6 degrees, and 1.9 m and 8.7 degrees respectively.

Architecture	Median Error	β value
VGG-Based	(3.1/m, 18.6/degree)	105
ResNet-Based	(1.9/m, 8.7/degree)	85

Table 2. Test results with real range frames

As shown in table 1 and table 2, the trained networks can achieve higher accuracy with synthetic test frames than with real lidar frames. The factor causing the higher error is the difference between the real lidar data and the generated synthetic image. Figure 8 shows a pair of real lidar range image and a synthetic range image generated with the same pose as the real lidar range image. We reshaped the 32×2160 range images to a 256×270 range images for easier observation. The differences come from the following three resources: (1). Noises and missing points are in real lidar data due to low laser intensity and transparent surfaces; (2). The real environment is more complex and contains more geometric details than the 3D model; (3). The 3D model is incomplete and might contain structural differences with respect to the actual indoor environment.

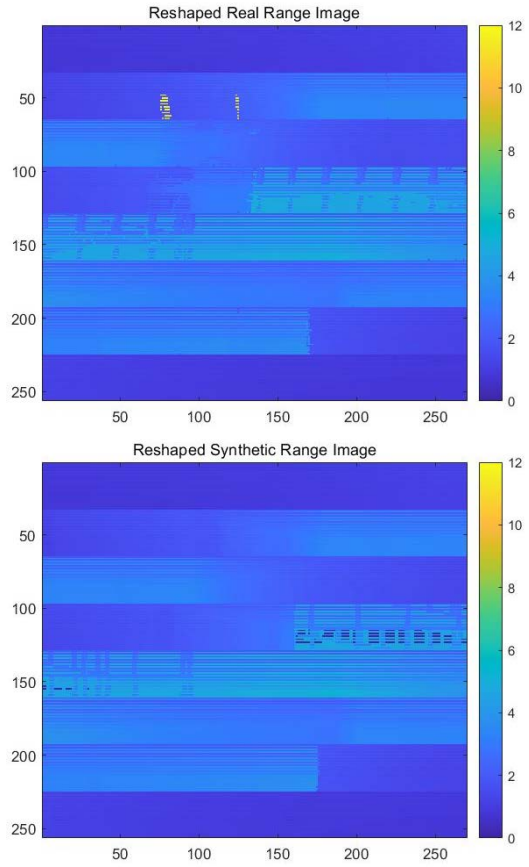

Figure 8. Reshaped real range image and synthetic range image

As figure 8 shows, most pixels in the real range image and the synthetic images are similar but there are still some points are

quite different. These large differences are caused by the complex real environment and the incomplete parts of the 3D model. The yellow points, representing large values, in the real range image are caused by windows. The laser fired by the Velodyne lidar goes through the window and hit a further wall, and then a large distance value is returned. The black points, representing zero values, in the synthetic range image is caused by the missing planes in the 3D model. The simulated lidar fires a ray in a certain direction, but it cannot hit any wall and returns a zero value.

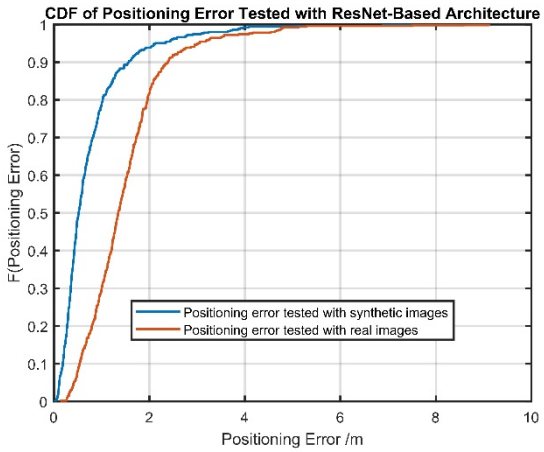

Figure 9. CDF of positioning errors tested with the ResNet-based CNN regression network

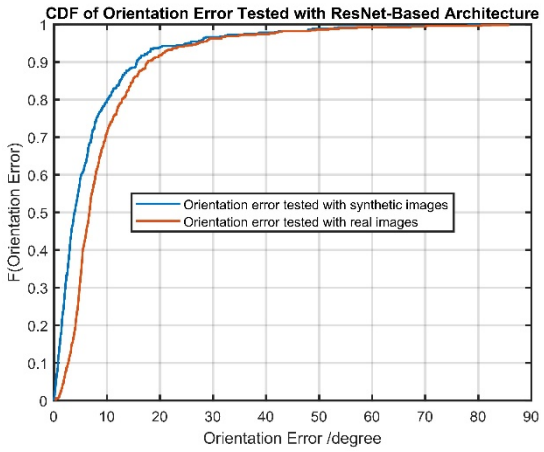

Figure 10. CDF of Orientation errors tested with the ResNet-based CNN regression network

Figure 9 and figure 10 show the CDF curves of the positioning errors and orientation errors tested with the ResNet-based CNN regression network.

5. CONCLUSION

This paper proposed a lidar relocalization method based on a CNN regression network using a 3D model. Synthetic training and test range images are generated from the 3D model. Experiments are conducted to select a CNN architecture and a suitable β value that can perform well with the input synthetic training and test range frames. Real lidar range images are input into the trained CNN regression network to evaluate the accuracy of the estimated pose vectors. Finally, we compared the results with synthetic test range images and real lidar range images, and analysed the reason why testing with real lidar images achieves a lower accuracy than testing with synthetic range images. Future works include training and testing the CNN regression networks in more complex indoor scenes.

REFERENCES

Acharya, D.,Khoshelham, K.,Winter, S, 2019a. BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. ISPRS Journal of Photogrammetry Remote Sensing**,** 150(4), 245-258.

Acharya, D., Singha Roy, S., Khoshelham, K. and Winter, S., 2019b. Modelling uncertainty of single image indoor localisation using a 3D model and deep learning. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, IV-2/W5, 247–254.

Caron, G.,Dame, A.,Marchand, E, 2014. Direct model based visual tracking and pose estimation using mutual information. Image Vision Computing**,** 32(1), 54-63.

Cheng, L.,Chen, S.,Liu, X.,Xu, H.,Wu, Y.,Li, M.,Chen, Y, 2018. Registration of laser scanning point clouds: A review. Sensors**,** 18(5), 1641.

Dubé, R.,Cramariuc, A.,Dugas, D.,Nieto, J.,Siegwart, R.,Cadena, C, 2018. SegMap: 3D segment mapping using data-driven descriptors. arXiv preprint arXiv:.09557.

Fischler, M. A.,Bolles, R. C, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM**,** 24(6), 381-395.

He, K.,Zhang, X.,Ren, S.,Sun, J, 2016. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.

Kendall, A.,Cipolla, R, 2017. Geometric loss functions for camera pose regression with deep learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5974-5983.

Kendall, A.,Grimes, M.,Cipolla, R, 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. Proceedings of the IEEE international conference on computer vision, 2938-2946.

Khoshelham, K.,Vilariño, L. D.,Peter, M.,Kang, Z.,Acharya, D, 2017. The ISPRS benchmark on indoor modelling. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-2/W7, 367–372.

Mur-Artal, R.,Tardós, J. D, 2017. Visual-inertial monocular SLAM with map reuse. IEEE Robotics Automation Letters**,** 2(2), 796-803.

Özuysal, M.,Lepetit, V.,Fleuret, F.,Fua, P, 2006. Feature harvesting for tracking-by-detection. European conference on computer vision, Springer, 592-605.

Reitmayr, G.,Drummond, T, 2006. Going out: robust model-based tracking for outdoor augmented reality. ISMAR, 109-118.

Saeedi, S.,Trentini, M.,Seto, M.,Li, H, 2016. Multiple-robot simultaneous localization and mapping: A review. Journal of Field Robotics**,** 33(1), 3-46.

Schiotka, A.,Suger, B.,Burgard, W, 2017. Robot localization with sparse scan-based maps. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 642-647.

Shotton, J.,Glocker, B.,Zach, C.,Izadi, S.,Criminisi, A.,Fitzgibbon, A, 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2930-2937.

Szegedy, C.,Ioffe, S.,Vanhoucke, V.,Alemi, A. A, 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. Thirty-first AAAI conference on artificial intelligence, 4278-4284.

Tian, Q.,Gao, Y.,Li, G.,Song, J, 2019. A Novel Global Relocalization Method Based on Hierarchical Registration of 3D Point Cloud Map for Mobile Robot. 2019 5th International Conference on Control, Automation and Robotics (ICCAR), 68-73.

Wang, L.,Li, R.,Zhao, L.,Hou, Z.,Li, X.,Sun, Z, 2017. Research on service robots robust relocalization algorithm based on 2D/3D map of indoor environment. 2017 18th International Conference on Advanced Robotics (ICAR), 572-577.

Williams, B.,Klein, G.,Reid, I, 2007. Real-time SLAM relocalisation. 2007 IEEE 11th International Conference on Computer Vision, 1-8.

Zhang, M.,Chen, Y.,Li, M, 2019. SDF-Loc: Signed Distance Field based 2D Relocalization and Map Update in Dynamic Environments. 2019 American Control Conference (ACC), 1997-2004.

Zhang, X.,Zhang, A.,Wang, Z.,Progress, O, 2014. Point cloud registration based on improved normal distribution transform algorithm. Laser Optoelectronics Progress**,** 51(4), 96-105.